# Dialogue-RAG: Enhancing Retrieval for LLMs via Node-Linking Utterance Rewriting

**Qiwei Li[1†], Teng Xiao[1†], Zuchao Li[1*], Ping Wang[2], Mengjia Shen[3], Hai Zhao[4]**
[1]School of Computer Science, Wuhan University, Wuhan, China,
[2]School of Information Management, Wuhan University, Wuhan, China,
[3]Wuhan Second Ship Design and Research Institute, Wuhan, China,
[4]School of Computer Science, Shanghai Jiao Tong University, Shanghai, China
{qw-line, xiaoxiao, zcli-charlie, wangping}@whu.edu.cn,
shenmj13@163.com, zhaohai@cs.sjtu.edu.cn

## Abstract

Large Language Models (LLMs) and Retrieval Augmented Generation (RAG) methods have demonstrated significant potential on tasks across multiple domains. However, ellipses and coreferences, as common phenomena in dialogue scenes, pose challenges to LLMs' understanding and RAG's retrieval accuracy. The previous works ignore the negative impact of this fuzzy data on RAG system. We explore the capabilities of LLMs and RAG systems in dialogue scenarios and use Incomplete Utterance Rewriting (IUR) to complete the key information in dialogue to enhance retrieval. Besides, we propose a lightweight IUR model for query rewriting. It is an end-to-end framework for node linking and iterative inference, incorporating two newly proposed probing semantic features derived from generative pre-training. This framework treats IUR as a series of link decisions on the input sequence and the incrementally constructed rewriting outputs. To test the performance of RAG system in the model multi-round dialogue scenario, we construct an RAG dialogue dataset on English and Chinese, Dialogue-RAG-MULTI-v1.0. Experiment results show that utterance rewriting can effectively improve the retrieval and generation ability of RAG system in dialogue scenes. Experiments on IUR tasks demonstrate the excellent performance of our lightweight IUR method.

## 1 Introduction

Human-machine interactions are ubiquitous in today's international web age. With continuous technological advancements, especially in deep learning, machines have gained the ability to understand human speech and language (Dongbo et al., 2023).

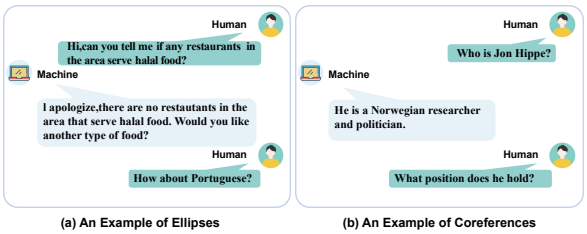


Figure 1: Two examples in human-machine dialogue. The question "How about Portuguese?" in Example (a) omits the noun "food". The pronoun "he" in the second question of Example (b) is a reference to "Jon Hippe".

Recently, the emergence of Large Language Models (LLMs) (DeepSeek-AI, 2024; Yang et al., 2024; GLM et al., 2024; Li et al., 2023c; Yao et al., 2024) and Retrieval Augmented Generation (RAG) (Ram et al., 2023; Asai et al., 2023a; Shi et al., 2023; Gao et al., 2023a; Asai et al., 2023b; Cuconasu et al., 2024) technologies has brought new opportunities for the development of human-computer dialogue.

Retrieval Augmented Generation is a technique that enhances text generation by utilizing information from additional data sources. In the context of LLMs, RAG technology has demonstrated outstanding performance in improving generation tasks. On one hand, models can use retrieved information to assist in generation, enhancing the quality and relevance of the generated text. On the other hand, RAG helps models acquire external knowledge and information, significantly expanding application scenarios of large language models. (Gao et al., 2023b; Gupta et al., 2024)

However, in real-world human-machine interaction scenarios, ellipses and coreferences occur quite frequently. Figure 1 illustrates an example of this situation. In dialogue tasks, the ambiguous information can hinder the ability of LLMs to accurately understand the semantics of user queries. In a RAG system, this ambiguity also impacts the retrieval of key information. The inaccurate retrieval results will lead to a decline in the generation per-

*Corresponding author.
†Equal contribution.

This work was supported by the National Natural Science Foundation of China (No. 62306216), the Technology Innovation Program of Hubei Province (No. 2024BAB043) and the National Social Science Fund of China (No. 24&ZD186).



*Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24423–24438
July 27 - August 1, 2025 

formance of LLMs.

To address this limitation, we propose a retrieval enhancement method for RAG, Dialogue-RAG, based on Incomplete Utterance Rewriting (IUR). It completes utterance using IUR, and realizes accurate RAG retrieval and model generation by more complete query. Besides, a novel lightweight IUR model is designed to complete the sentence information to improve the quality of the search results, and then improve the question-answering ability of the generated model. To evaluate the ability of RAG in the dialogue scenario, we propose an RAG dialogue dataset on English and Chinese, which constructs a question answering pair in the scenario of knowledge dialogue and a knowledge base for retrieving related information. This can effectively test the knowledge retrieval and question answering ability of RAG method in dialogue scenario.

Our main contributions are as follows:

- We explore the dialogue task scenario and design a new retrieval enhancement method for RAG based on IUR. We build an enhanced RAG system based on this approach (Dialogue-RAG) and it can achieve accurate retrieval of knowledge base by rewriting and completing incomplete data. Furthermore, it effectively improve the model's ability to answer fuzzy questions in real scenarios.

- We design a lightweight IUR model that transforms the IUR task into a digraph parsing task and uses a node linking parser and generative pre-trained language model to realize fast digraph parsing decision-making in the IUR tasks.

- We manually construct a retrieval-augmented generation dialogue dataset on English and Chinese, which contains multiple rounds of question answering pairs and a knowledge base for retrieval. It can not only test the dialogue ability of the model, but also evaluate the effect of RAG method on the enhancement of dialogue ability.

## 2 Related Work

### 2.1 Retrieval Augmented Generation

Retrieval-Augmented Generation enhances language models by integrating retrieved text passages, significantly improving performance on knowledge-intensive tasks. Initial work by (Guu et al., 2020) and (Lewis et al., 2020) combined dense passage retrieval with sequence-to-sequence models, showing notable performance gains (Ram et al., 2023). Recent advancements include (Luo et al., 2023), who instruction-tuned models with fixed retrieved passages and pre-trained retrievers and models together. (Jiang et al., 2023) proposed adaptive retrieval during generation, and Toolformer (Schick et al., 2024) trained models to generate API calls. However, these approaches face challenges in efficiency, handling irrelevant context, and attribution accuracy (Mallen et al., 2023; Shi et al., 2023; Liu et al., 2023a). Self-RAG (Asai et al., 2023b) addresses these issues by allowing models to use retrieval on demand for diverse queries to some extent. (Cuconasu et al., 2024) studied the impact of several key factors, like the type, number, and position of documents that should augment the prompt to the LLM. However, the existing benchmarks ignore the negative effects of ellipsis and coreference on knowledge retrieval.

### 2.2 Incomplete Utterance Rewriting

Early works on Incomplete Utterance Rewriting primarily employ sequence-to-sequence models combined with a copy mechanism to extract relevant information from the context (Su et al., 2019; Elgohary et al., 2019; Kumar and Joshi, 2017). These methods use pointer networks or sequence generation models to extract information and address basic rewriting issues. With the introduction of pre-trained models, Incomplete Utterance Rewriting (IUR) research has significantly advanced. For example, RUN (Liu et al., 2020a) proposes a rewriting matrix based on context and utterance embeddings, which determines whether insertion or replacement operations should be carried out by annotating each token. Additionally, SRL (Xu et al., 2020) pre-identifies the subject, predicate, and object within the context and incorporates these as extra features into the encoder, improving performance. RaST (Hao et al., 2021) models the IUR problem as a sequence labeling task, implementing rewriting by predicting actions (insertion, deletion, and None) and the spans of the context separately. HCT (Jin et al., 2022) further enhances the action predictor from previous work to a rule predictor, optimizing the generation of modified words not explicitly present in the context. Recently, (Li et al., 2023b) improve IUR performance by continued pre-training of the T5 (Raffel et al., 2020) model. Meanwhile, (Du et al., 2023) and (Li et al., 2023a)

further develop fine-grained subtasks and refined model architectures based on the edit matrix. Considering that the context of the dialogue contains similar content that tends to cause confusion and the contents of the different themes are numerous and complex, (Guo et al., 2024) design the Dynamic Context Introduction mechanism to filter out irrelevant contexts to handle the extended multi-turn dialogue. XSS (Peng et al., 2024) designs the Cross Scorer Sharing mechanism to support efficient pair locating for IUR, enabling faster and more accurate processing of extended dialogue contexts.

## 3 Methods

### 3.1 Dialogue Retrieval Augmented Generation

To effectively deal with the degradation of retrieval quality caused by incomplete utterance, we design a **Dialogue Retrieval Augmented Generation (Dialogue-RAG)** method. The overall architecture of the method is shown in Figure 2. We rewrite the input question according to the history context and complete the missing parts. In RAG processing, the completed information is used for corresponding information retrieval and question answering. In this way, more accurate information retrieval and high-quality question answers are achieved.

Specifically, suppose the multi-turn dialogue utterances are $U = \{u^1, u^2, ..., u^n\}$. We use our new lightweight IUR model rewrite utterance $u^i$ to a self-contained version $Ru^i$ with context $U_c = \{u^1, u^2, ..., u^{i-1}\}$, where $u^i = \{w_1^i, w_2^i, ..., w_m^i\}$ is the $i$-th utterance with $m$ words in dialogue. We define the procedure as follows:

$$Ru^i = IUR(\{u^1, u^2, ..., u^{i-1}\}, u^i), \quad (1)$$

where $IUR(\cdot)$ is the rewriting process. Details are provided in Subsection 3.2 of this chapter. Based on $Ru^i$, we match key information through similarity. For knowledge base information $V = \{v_1, v_2, ..., v_l\}$, $v_i$ represents the chunk vector in the knowledge base. We use cosine similarity to calculate the similarity $s_{i,j}$ between $Ru^i$ and $v_j$:

$$s_{i,j} = \frac{\mathbf{Ru}^i \cdot \mathbf{v}_j}{\|\mathbf{Ru}^i\| \|\mathbf{v}_j\|}. \quad (2)$$

We reorder chunks according to the score $s_{i,j}$ and select the $Top-k$ chunks as the matching information $T_m = \{t_1, t_2, ..., t_k\}$ for $Ru^i$. The $t_j \in T_m$ represents the chunk text corresponding to the chunk vector $v_j$. As shown in Figure 2, matched chunk information $T_m$ and rewritten utterance $Ru^i$ will be fused into the prompt template as a context prompt $P(T_m, Ru^i)$ and it will be used as input to the large language model for the dialogue answer $a_i$:

$$a_i = \text{LLM}(P(T_m, Ru^i)), \quad (3)$$

where $\text{LLM}(\cdot)$ is the large language model.

### 3.2 Node Linking Iterative Inference

To meet the retrieval requirements of RAG, we design a lightweight IUR model. Following the practice of previous text editing works (Liu et al., 2020a; Zhang et al., 2022), context utterances $U_c$ are concatenated into a single word sequence $S_c$ with a special token `[SEP]` as the utterance separator. In this part, there are three components to our model: a node linking parser, a contextual similarity feature (CSF) extractor, and a rewrite consistency feature (RCF) extractor. The CSF and RCF feature extractors are GPT model-based probes to enhance the interactive inference in IUR.

**Digraph Modeling.** For IUR, it mainly uses the determined phrase from the context and inserts it into a certain position of the incomplete utterance. We transform it into a digraph modeling task and regard the beginning and ending words of the phrase span in the context and the words on the left of the insertion position in the incomplete utterance as nodes. The edge between two nodes acts as a link. We define two types of links: span link and insertion link. The first is used to determine the span of the phrase to be inserted in the context, while the second is used to determine the insertion position in the incomplete utterance.

It is worth noting that although insertion can solve the vast majority of rewriting problems, there are still some words in the rewritten sentence that don't come from the context. Since these words do not depend on context, an additional model such as grammatical error correction can be used for post-processing. However, this phenomenon is not the focus of this paper and is therefore not considered in our digraph modeling transformation.

Digraph modeling can effectively simplify the IUR problem. Since for a context of length $m$, an incomplete utterance of $n$, the complexity of the insertion operation is $O(m \times n)$ and the complexity for determining the insertion span is $O(m \times m)$, so the overall complexity is $O(m \times m \times n)$. Digraph

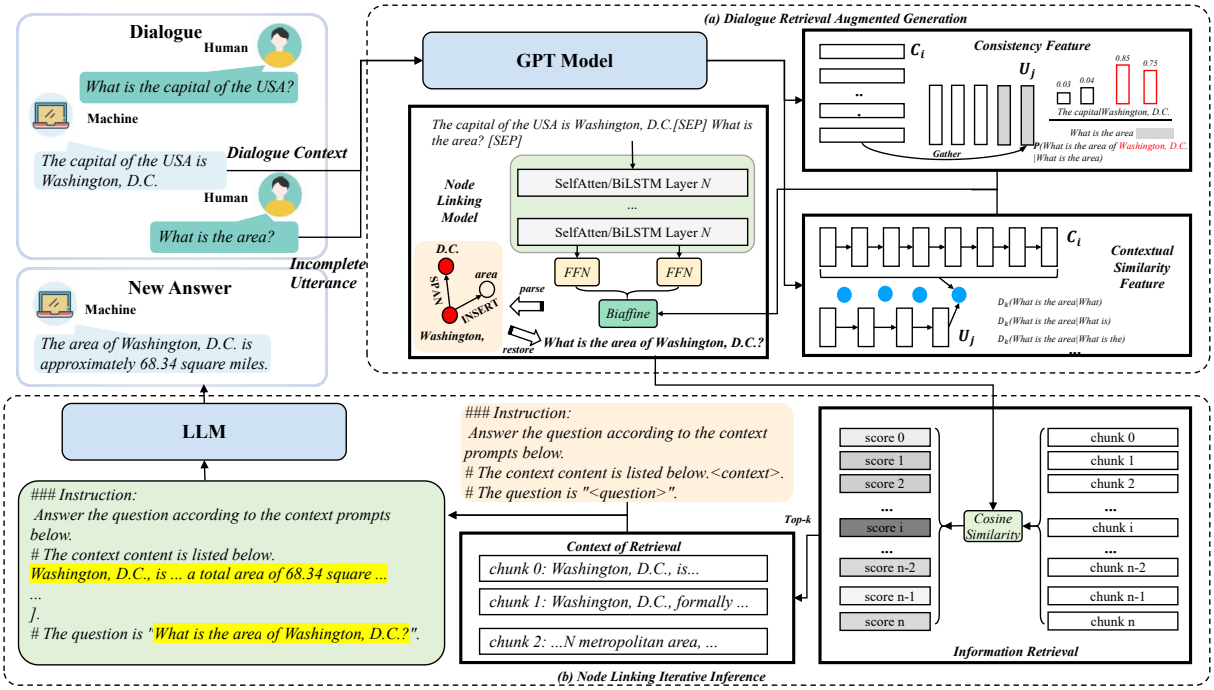


Figure 2: Overall architecture.

modeling decomposes it into two steps of digraph parsing, determining the insertion point has complexity $O(m \times n)$, and determining the span has complexity $O(m \times m)$, so the overall implementation complexity is $O(m \times n) + O(m \times m)$, which greatly reduces inference difficulty. In the actual implementation, the two-step link inference is finished at one time, so the final complexity is reduced to $O\big((m+n) \times (m+n)\big)$.

**Node Linking Parser.** Based on digraph modeling, we propose to use a node linking parser to learn the digraphs. The overall structure of the parser is shown on the left side of Figure 2. Formally, for an input sequence $X = \{w_1, w_2, ..., w_n\}$ with length $n$ (which is a concatenation of context and incomplete utterances), we insert a special token `BOS` at the beginning of the sequence. We employ a sequence encoder to convert the input sentence into contextualized vector representations $H$.

In order to distinguish the tokens in the sequence as the head node and the tail node of the link, we introduce two independent feed-forward neural (FFN) layers to project the representations from the sequence encoder to different spaces respectively:

$$\begin{aligned} H^{(head)} &= \text{FFN}_{\text{head}}(H), \\ H^{(tail)} &= \text{FFN}_{\text{tail}}(H). \end{aligned} \tag{4}$$

To capture the link relationship between the head node and the tail node, we introduce biaffine attention scorer (Dozat, 2016) to score the relationship between the head-tail node pair.

$$s(i \xrightarrow{l} j) = H_i^{(head)\top} W H_j^{(tail)} + b, \tag{5}$$

where $i$ and $j$ indicate the index of the head and tail node in the sequence respectively, $W$ and $b$ are learnable parameters, and $l$ is the relation label of link $i \rightarrow j$. A multi-class cross-entropy loss is applied to the biaffine score $s(i \xrightarrow{l} j)$ to guide the model to learn node linking parsing.

$$\mathcal{L} = \sum_{i,j} XEnt(s(i \xrightarrow{l} j), r(i \rightarrow j)), \tag{6}$$

where $r(i \rightarrow j)$ indicates the gold relation label between node $i$ and $j$, and `NONE` is used to indicate that there is no relationship between nodes.

**Iterative Inference.** The foundation of our proposed two probing features is generative language modeling. Its modeling objective for the input language sequence $X = \{x_1, x_2, ..., x_L\}$ is to predict each token $x_i$ using the unidirectional visible token sequence $X_{<i}$ and maximize the following maximum likelihood so that the model can learn the language generation knowledge in an unsupervised way. GPT (Liu et al., 2023b), which is widely used in natural language generation and other fields, is one of many models that have been pre-trained with generative language modeling, thus becoming the

basis for our probing feature. Formally, the probability of distribution for the next token prediction in GPT predicting is given by:

$$P_{\mathcal{G}}(y) = \textbf{GPT}(x_0, ..., x_{i-1}). \tag{7}$$

where $y$ represents the next predicted token.

*a. Contextual Similarity Feature.*

In the IUR inference process, assuming incomplete utterance rewriting depends on the phrase $u_{k:k+lambda} = \{w^u_k, w^u_{k+1}, \ldots, w^u_{k+\lambda}\}$ in the context utterance $u$. $u_{k:k+lambda}$ is a complement to incomplete utterance, and the complement insertion position is $p$. From the perspective of generative language modeling, the generation context of phrase $u_{k:k+lambda}$ in original utterance is $u_{<k} = \{w^u_1, w^u_2, ..., w^u_{k-1}\}$ while the generation context at the insertion position $p$ incomplete utterance is $iu_{<p}$. Since two generation context aims at generating the same phrase $u_{k:k+lambda}$, we refer to this case as contextual similarity assumption. That is, by the context $u_{<k}$ and $iu_{<p}$ has similarity in the distribution of the next token prediction.

Formally, for the generation context $u^i_{<k}$ and $iu_{<p}$, GPT model output the next token distribution respectively as:

$$\begin{aligned} P_{\mathcal{G}}(y)^u &= \textbf{GPT}(w^u_1, w^u_2, ..., w^u_{k-1}), \\ P_{\mathcal{G}}(y)^{iu} &= \textbf{GPT}(w^{iu}_1, w^{iu}_2, ..., w^{iu}_{p-1}). \end{aligned} \tag{8}$$

According to the contextual similarity assumption, the divergence between the two distributions should be smaller if the token from the $k$-th position of the utterance $u$ can be inserted into the position $p$ of the incomplete utterance $iu$. In other words, the distribution divergence measures the extent to which a token at a specific position in the utterance can be inserted into the incomplete utterance in IUR. Thus, the contextual similarity feature (CSF) can be written as:

$$\begin{aligned} CSF =& D_{KL}(P_{\mathcal{G}}(y)^{iu} || P_{\mathcal{G}}(y)^u) \\ =& \sum P_{\mathcal{G}}(y)^{iu} \log \frac{P_{\mathcal{G}}(y)^{iu}}{P_{\mathcal{G}}(y)^u}. \end{aligned} \tag{9}$$

*b. Rewrite Consistency Feature.*

Apart from context similarity feature, for the typical scenario of incomplete utterance rewriting, inserting the phrase $u_{k:k+\lambda} = \{w^u_k, w^u_{k+1}, ..., w^u_{k+\lambda}\}$ started with position $k$ in the context into the position $p$ of the incomplete utterance should make the new utterance a reasonable sentence. In other words, the partial sequence of the incomplete utterance $iu_{\leq p}$ and the inserted word $u_k$ will form a new valid sentence fragment. We refer to such phenomenon as rewriting consistency.

Based on this phenomenon, we deduce a rewrite consistency feature (RCF) to enhance the determination of completion words and insertion positions in IUR. Since GPT's pre-training goal is to predict the next token, we can evaluate the probability of this prediction as the likelihood that the token and its context form a valid segment. In IUR, we adopt the probability value of a given token from GPT prediction distribution as the rewrite consistency score, which reflects the probability that the partial sequence $iu_{\leq p}$ and $u_k$ to form a valid fragment. For all words in the context, a vector composed of scores is calculated accordingly to obtain the rewrite consistency feature for insertion position $p$. Formally, for position $p$ in the incomplete utterance and utterance context $u = \{w^u_1, w^u_2, ..., w^u_m\}$, the rewrite consistency feature is written as:

$$RCF = [\textbf{GPT}(w^u_k | w^{iu}_1, w^{iu}_2, ..., w^{iu}_p)]^m_{k=1}, \tag{10}$$

where $[\cdot]^m_{k=1}$ represents a list operation from 1 to $m$ elements.

*c. Feature Integration.*

Denote that context length is $m$ and incomplete utterance length is $n$. In the CSF calculation, any two token positions in the incomplete utterance and context are probed to obtain a divergence score, then the final CSF feature size is $m \times n$. Similarly, the RCF feature size is also $m \times n$. Since in node linking parsing, we concatenate context and incomplete utterance into one sequence. Therefore, we extend the feature to $(m+n) \times (m+n)$ by zero padding to facilitate the use of the feature. The feature integration is achieved by adding the padded features to the biaffine score, i.e.,

$$\begin{aligned} s(i \xrightarrow{l} j) = & H^{(head)\top}_i W H^{(tail)}_j \\ & + CSF_{i,j} + RCF_{i,j} + b, \end{aligned}$$

where b is the bias.

### 3.3 Dialogue RAG Dataset

To test the performance of LLM and RAG methods in real dialogue scenarios, we manually construct an dialogue dataset on English and Chinese, **Dialogue-RAG-MULTI-v1.0**. The dataset provides multiple rounds of conversations as well as related document. We crawl web pages to get the documents and converted them into serialized text for retrieval knowledge bases. For the test dataset, we

| Spilt | Dialogue Rounds | Q&A Number | Document Number |
|---|---|---|---|
| Train | Total: 23,996<br>English: 12,005<br>Chinese: 11,991 | Total: 226,173<br>English: 124,852<br>Chinese: 101,321 | Total: 9,321<br>English: 4,889<br>Chinese: 4,432 |
| Dev | Total: 762<br>English: 391<br>Chinese: 371 | Total: 6,310<br>English: 3,180<br>Chinese: 3,130 | |
| Test | Total: 1,036<br>English: 542<br>Chinese: 494 | Total: 8,216<br>English: 4,321<br>Chinese: 3,895 | |

Table 1: Dataset setting.

use the document to annotate the conversation data in the form of manual annotations. For the training and development datasets, annotations were generated using DeepSeek-V3's API, followed by manual review to remove low-quality data. The conversations are constructed based on the web pages. Dialogue-RAG-MULTI-v1.0 provides subjects with a real complex dialogue scene, which contains special concepts, contextual reference, incomplete utterance and other issues that often occur in daily dialogue process.

In practical application scenarios, we can not only directly test the multi-round dialogue effect of the model, but also evaluate the dialogue ability of the model enhanced by the RAG method. The distribution of the dataset is shown in Table 1. According to the statistics of test dataset, the proportion of ellipses and coreferences in multi-round conversations exceeds 60%, which can effectively test RAG system in dialogue scenarios.

## 4 Experiments

### 4.1 Setup

**Dataset.** We conduct knowledge dialogue task on our proposed Dialogue-RAG-MULTI-v1.0 dataset to test the effect of rewriting in RAG augmented human-machine dialogue tasks. To further explore the impact of utterance rewriting on retrieval, we conduct retrieval comparison on the CoQA (Reddy et al., 2019), TOPIOCQA (Adlakha et al., 2022) and OR-QuAC (Qu et al., 2020) dataset. To evaluate the validity of our IUR component, we conduct experiments on English Task-Oriented Dialogue dataset CANARD (Liu et al., 2020b) and Chinese Open Domain Dialogue dataset REWRITE (Su et al., 2019). Same dataset split as in the original paper has been adopted to ensure reproducibility.

**Baselines for RAG.** In order to verify the effect of our method, we test the knowledge dialogue task in the condition of the generation without IUR and RAG (w/o IUR), the generation without RAG (w/ IUR), the generation without IUR (RAG w/o IUR), the generation of IUR with few shot based on GPT-4o-mini (RAG w/ GPT-4o-mini few shot IUR), and the generation of IUR with our Node Linking Iterative (Dialogue-RAG). In addition, we also compare the performance with other RAG methods, including GLM-RAG (ZhipuAI, 2024), AAR (Yu et al., 2023), Adaptive-RAG (Jeong et al., 2024), IRCot (Trivedi et al., 2022) and DRAGON+ (Liu et al., 2024).

**Baselines for IUR** For the IUR task, in the setting without BERT pre-trained language model enhancement, the LSTM encoder is used in the baselines Syntactic, L-Gen, L-Ptr, and L-Ptr-Gen, which are consistent with our work. And vanilla attention-based generator (Bahdanau et al., 2015), pointer network generator (Vinyals et al., 2015) and hybrid pointer generator (See et al., 2017) is employed in L-Gen, L-Ptr, and L-Ptr-Gen, respectively. For DuS (Kumar and Joshi, 2016), a semantic sequence and a syntactic sequence model are combined to generate accurate rewrites. And RUN (Liu et al., 2020b) shaped the problem as the prediction of a word-level edit matrix and proposed a semantic segmentation model (UNet) to predict the edit operations. For the comparison with LLMs, we choose the open source models Qwen2-7B-Instruct (Yang et al., 2024) and ChatGLM3-6B (GLM et al., 2024). In addition, we tested DeepSeek-V3 (DeepSeek-AI, 2024) and GPT-4o-mini (OpenAI, 2024). For the REWRITE dataset, our baseline Transformer-as-encoder versions (T-Gen, T-Ptr and T-Ptr-Gen) are used to analyze the performance difference brought by the encoder architecture. L-Ptr-$\lambda$ and T-Ptr-$\lambda$ are variants of a pointer network that use $\lambda$ to control whether to copy tokens from context or from utterance, instead of deciding to copy or generate as in the traditional pointer network.

**Evaluation Metrics.** We use BLEU, ROUGE, and BERTScore automatic metrics for RAG enhanced dialogue scene. In the retrieval experiment, we use accuracy, recall and MRR@5 to evaluate the retrieval effect. Besides, we use BLEU, ROUGE, and EM automatic metrics for IUR performance evaluation as done in (Pan et al., 2019)

**Model Settings.** For the knowledge dialogue task, `ChatGLM3-6B` (GLM et al., 2024)

| Method | $B_1$ | $B_2$ | $B_3$ | $B_4$ | $\mathcal{R}_1$ | $\mathcal{R}_2$ | $\mathcal{R}_L$ | $\mathcal{S}_p$ | $\mathcal{S}_r$ | $\mathcal{S}_f$ |
|---|---|---|---|---|---|---|---|---|---|---|
| GLM-RAG | 8.00 | 5.33 | 4.00 | 3.06 | 16.83 | 8.05 | 15.73 | 55.74 | 66.47 | 60.16 |
| AAR | 8.78 | 5.88 | 4.43 | 3.33 | 14.35 | 7.29 | 13.28 | 52.07 | 60.78 | 55.80 |
| Adaptive-RAG | 7.33 | 4.90 | 3.67 | 2.76 | 15.56 | 7.05 | 14.62 | 56.61 | 63.01 | 58.51 |
| IRCot | 5.43 | 3.58 | 2.68 | 1.99 | 13.04 | 5.65 | 12.25 | 56.39 | 58.34 | 55.67 |
| DRAGON+ | 11.91 | 8.73 | 6.88 | 5.41 | 21.53 | 12.38 | 20.07 | 56.37 | 68.4 | 61.46 |
| w/o IUR | 5.84 | 3.31 | 2.25 | 1.58 | 10.34 | 3.5 | 9.55 | 52.82 | 57.37 | 54.83 |
| w/ IUR | 6.35 | 3.53 | 2.36 | 1.61 | 11.13 | 3.64 | 10.2 | 52.82 | 59.75 | 55.9 |
| RAG w/o IUR | 10.29 | 7.18 | 5.46 | 4.14 | 19.04 | 9.82 | 17.91 | 57.48 | 67.26 | 61.62 |
| RAG w/ GPT-4o-mini few shot IUR | 13.03 | 9.59 | 7.49 | 5.79 | 25.86 | 14.57 | 24.41 | 59.82 | 73.91 | 65.81 |
| **Dialogue-RAG (ours)** | **13.11** | **9.69** | **7.63** | **5.95** | **26.09** | **14.72** | **24.6** | **60.33** | **74.09** | **66.21** |

Table 2: Main results. $B$, $R$, and $S$ are BLEU, ROUGE, and BERT SCORE respectively. The cumulative n-gram BLEU score is denote as $\boldsymbol{B_n}$. For ROUGE metric, it measures the n-gram overlapping (denoted as $\boldsymbol{R_n}$) and longest matching (denoted as $\boldsymbol{R_L}$) between the rewritten utterances and the golden ones. $\mathcal{S}_p$, $\mathcal{S}_r$, and $\mathcal{S}_f$ represent the precision, recall, and F1 score of BERT SCORE respectively.

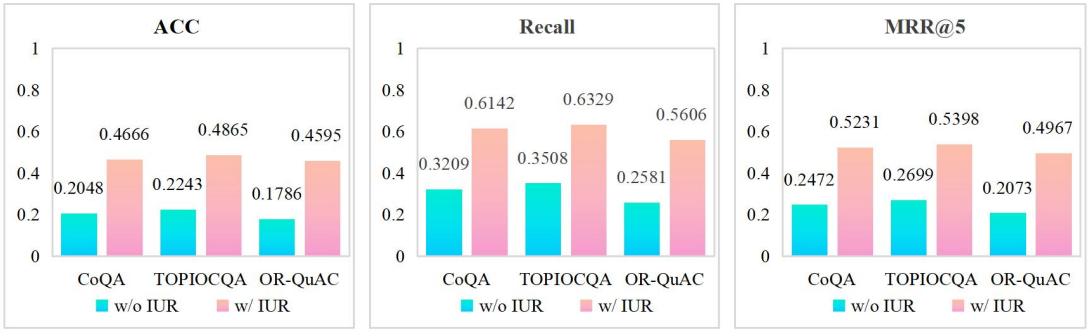


Figure 3: Comparison of results with and without rewrite on test sets across three datasets

is used as our inference model. We test `Qwen2.5-7B-Instruct` (Team, 2024) and `Llama3-8B-Chinese-Chat` (Wang et al., 2024) as the base model and show their results in Appendix A. We test BiLSTM and Transformer as the IUR module encoder. The encoder setting can be found in Appendix B. For CSF and RCF feature extraction, we use the official `GPT2-base`. For the IUR experiment of LLM, we set the number of the LLM few shot samples as 3. For the non-LLM model, we tested it on a GeForce RTX 3090 graphics card. For LLM, we deployed models on two GeForce RTX 3090 GPUs for testing.

### 4.2 Main Results and Analysis

We test the results of different RAG systems on the incomplete utterance dialogue dataset. As shown in Table 2, we can find that the results become better after IUR (w/ IUR) or RAG (RAG w/o IUR) augmentation, which verifies the positive effect of IUR and RAG method in incomplete utterance dialogues. Besides, We can find the method of using IUR to enhance RAG (Dialogue-RAG) achieves the best results in various indicators, which fully confirms the effect of our method.

To demonstrate the retrieval improvement of IUR, we separately compare the accuracy of correctly finding the dialogue document between direct retrieval (w/o IUR) and retrieval (w/ IUR) after rewriting. The results (w/o IUR, 37.80 vs w/ IUR, 77.54) show a surprising improvement in retrieval accuracy after rewriting incomplete sentences. In addition, we score the responses of the RAG system with GPT-4, and the results (w/o IUR, 4.89 vs w/ IUR, 7.70) also showed the gains brought by the IUR method.

In addition, we also test the impact of rewriting results generated using large language models on RAG. The results show that the results generated by our IUR model have better performance than the results generated by the large language model (RAG w/ GPT-4o-mini few shot IUR). This fully validates the excellent performance of our IUR method.

### 4.3 Retrieval Comparison

To further explore the retrieval effect of utterance rewriting in the dialogue field, we conduct retrieval

comparison on CoQA, TOPIOCQA and OR-QuAC datasets. The results are shown in Figure 3. The results on three datasets show that retrieval after rewriting is significantly better than direct retrieval. This shows that the utterance rewriting of the ambiguity in the dialogue by the IUR method can significantly improve the retrieval effect, which fully verifies our point.

### 4.4 Evaluation on IUR Task

To evaluate our IUR approach, we conducted comparative experiments on the CANARD and REWRITE benchmarks. Experiment results are presented in Tables 3 and 4. To show the effect of rewriting, we show the case comparison in Appendix C. Besides, We also show the results on other datasets in Appendix D.

For the CANARD dataset, we can find that our model obtains the best results on BLEU and ROUGE, which illustrates digraph modeling, CSF and RCF features are generally effective for IUR. Besides, we test some LLM baselines on the CANARD benchmark to compare our methods. From Table 3, we can see that LLMs with both zero shot and few shot perform worse than our method. It is important to note that our approach not only performs significantly better than large models. As a lightweight model, its computing speed is also much higher than LLM. This is very important for the retrieval efficiency of the RAG system.

For the REWRITE dataset, comparing the results of LSTM-based with Transformer-based baselines, we find that Transformer has an advantage in most systems, but not in all cases (L-Gen vs T-Gen), which may be due to the absence of optimization for Transformer. It suggests that just changing LSTMs with smaller parameters and training-data to Transformers may not necessarily be the best option, we keep using BiLSTM as the encoder when without BERT. On the REWRITE dataset, our full model still exhibits performance advantages with or without BERT. And since the REWRITE dataset is smaller, to increase the stability of the results, we use stricter EM and $\mathbf{B}_2$, $\mathbf{B}_4$, $\mathbf{R}_2$, $\mathbf{R}_L$ metrics. The reported improvements on these stricter metrics demonstrate the generalization ability of our method.

Additionally, we also test some LLM baselines on the CANARD benchmark to compare our methods. From Table 3, we can see that LLMs with both zero shot and few shot perform worse than our method. Besides, as a lightweight model, our IUR component achieves significantly higher computing speeds. This enhances the retrieval efficiency of RAG.

| Model | $\mathbf{B}_1$ | $\mathbf{B}_2$ | $\mathbf{B}_4$ | $\mathbf{R}_1$ | $\mathbf{R}_2$ | $\mathbf{R}_L$ |
|---|---|---|---|---|---|---|
| Traditional Model | | | | | | |
| L-Ptr | 52.4 | 46.7 | 37.8 | 72.7 | 54.9 | 68.5 |
| Pronoun Sub † | 60.4 | 55.3 | 47.4 | 73.1 | 63.7 | 73.9 |
| L-Ptr-Gen | 67.2 | 60.3 | 50.2 | 78.9 | 62.9 | 74.9 |
| RUN | 70.5 | 61.2 | 49.1 | 79.1 | 61.2 | 74.7 |
| Large Language Model w/ Zero shot | | | | | | |
| Qwen2 | 26.5 | 17.0 | 5.5 | 50.3 | 29.1 | 49.2 |
| ChatGLM | 33.6 | 24.8 | 12.2 | 58.9 | 40.8 | 54.8 |
| Deepseek | 27.1 | 18.6 | 7.5 | 50.4 | 31.5 | 46.8 |
| GPT-4o-mini | 34.6 | 25.9 | 18.6 | 60.2 | 42.5 | 56.6 |
| Large Language Model w/ Few shot | | | | | | |
| Qwen2 | 34.6 | 25.9 | 12.6 | 60.2 | 42.5 | 56.6 |
| ChatGLM | 36.9 | 28.1 | 14.7 | 62.8 | 45.0 | 59.3 |
| Deepseek | 42.6 | 34.8 | 20.9 | 70.4 | 55.3 | 67.8 |
| GPT-4o-mini | 44.4 | 36.7 | 22.2 | 72.8 | 58.0 | 70.5 |
| **Ours** | 71.5 | 62.8 | 51.3 | 81.0 | 63.5 | 76.0 |

Table 3: Experimental results on CANARD. †: Results from Liu et al. (2020b).

| Model | EM | $\mathbf{B}_2$ | $\mathbf{B}_4$ | $\mathbf{R}_2$ | $\mathbf{R}_L$ |
|---|---|---|---|---|---|
| L-Gen | 47.3 | 81.2 | 73.6 | 80.9 | 86.3 |
| L-Ptr-Gen | 50.5 | 82.9 | 75.4 | 83.8 | 87.8 |
| L-Ptr | 51.5 | 82.7 | 75.5 | 84.0 | 88.2 |
| L-Ptr-$\lambda$ † | 42.3 | 82.9 | 73.8 | 81.1 | 84.1 |
| T-Gen | 35.4 | 72.7 | 62.5 | 74.5 | 82.9 |
| T-Ptr-Gen | 53.1 | 84.4 | 77.6 | 85.0 | 89.1 |
| T-Ptr | 53.0 | 83.9 | 77.1 | 85.1 | 88.7 |
| T-Ptr-$\lambda$ | 52.6 | 85.6 | 78.1 | 85.0 | 89.0 |
| RUN | 53.8 | 86.1 | 79.4 | 85.1 | 89.5 |
| **Ours** | 57.6 | 86.7 | 79.8 | 85.3 | 90.8 |
| T-Ptr-$\lambda$ + BERT | 57.5 | 86.5 | 79.9 | 86.9 | 90.5 |
| RUN + BERT | 66.4 | 91.4 | 86.2 | 90.4 | 93.5 |
| **Ours + BERT** | 69.4 | 91.5 | 86.9 | 91.7 | 94.7 |

Table 4: Experimental results on REWRITE. †: Reproduced from the code released by Su et al. (2019).

### 4.5 Ablation Study

To illustrate the enhancement of CSF and RCF features on digraph modeling, we conduct an ablation study on the REWRITE dataset, and the results are shown in Table 5 and 6. We not only report the performance of IUR on the REWRITE dataset, but we also report the unique metrics of digraph modeling, parsing precision (PP), parsing recall (PR), parsing f-score (PF), and link triplet precision (TP), triplet recall (TR) and triplet f-score (TF). First, when the CSF or RCF feature is removed, the IUR and parsing performance both drop, indicating that these features can effectively help digraph decision-making. Among them, the CSF and RCF

| | EM | $B_2$ | $B_4$ | $R_2$ | $R_L$ |
|---|---|---|---|---|---|
| RUN + BERT | 66.4 | 91.4 | 86.2 | 90.4 | 93.5 |
| **Full Model** | 69.4 | 91.5 | 86.9 | 91.7 | 94.7 |
| w/o CSF | 69.0 | 90.9 | 86.3 | 90.8 | 94.1 |
| w/o RCF | 69.2 | 90.8 | 86.4 | 90.9 | 94.3 |
| w/o CSF, RCF | 68.6 | 90.9 | 86.3 | 90.8 | 94.1 |

Table 5: IUR performance on REWRITE test set.

| | PP | PR | PF | TP | TR | TF |
|---|---|---|---|---|---|---|
| **Full Model** | 82.6 | 76.8 | 79.6 | 76.9 | 70.1 | 73.3 |
| w/o CSF | 82.0 | 76.8 | 79.3 | 76.2 | 69.3 | 72.6 |
| w/o RCF | 81.7 | 77.4 | 79.5 | 75.7 | 70.6 | 73.1 |
| w/o CSF, RCF | 81.0 | 77.3 | 79.1 | 74.8 | 70.3 | 72.5 |

Table 6: Parsing performance on REWRITE test set.

features have the greatest impact on the EM metric, while the BLEU and ROUGE metrics are influenced slightly. Second, when CSF and RCF are completely removed, the results of digraph modeling still have a performance advantage compared to RUN, which shows the advantage of digraph modeling that simplifies the complexity without complex model designs. To further verify the advantages of our method, we conduct a case analysis in Appendix E.

### 4.6 Rewriting Speed

To illustrate the advantages of digraph modeling in IUR, we compare the rewriting speed between the RUN model, our base model, and our full model on the REWRITE test dataset in Figure 4. In order to minimize the influence of irrelevant factors, we run the three systems on the same machine for 5 times and report the average speed. Comparing the speed of RUN with our base model, it shows that modeling the IUR as a digraph structure is very efficient. Further use of our proposed CSF and RCF features reduces this efficiency advantage, but still achieves more than 3 times speedup. In addition, the speed of model w/ BERT is reduced compared with that w/o BERT, indicating that BERT encoding also occupies an important time overhead. In addition, the speed of LLM generation is about 20-30 tokens/s. We test the sentence generation speed of Qwen2-7B and ChatGLM3-6B on the same dataset and the results are 2.72 and 2.73 sentence per second, respectively. LLMs are far slower than our proposed method and not suitable for rewriting in RAG.

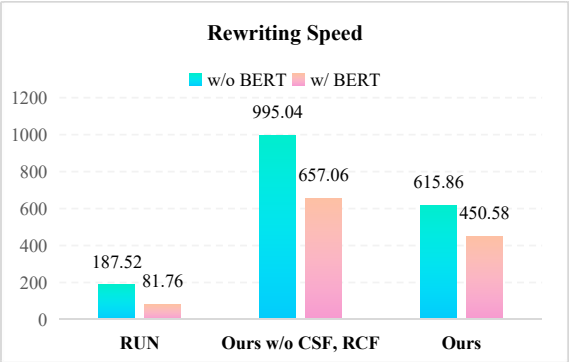


Figure 4: Comparison of rewriting speed (sentences per second) on REWRITE test dataset.

## 5 Conclusion

Ellipses and coreferences in dialogues limit the retrieval capabilities of RAG systems. We propose a RAG method based on IUR enhancement. This method uses the Node Linking IUR model to rewrite queries, enabling better retrieval. In addition, we design a knowledge conversation dataset to evaluate performance of RAG in real dialogue scenarios. The experimental results show that our approach achieves better results when dealing with incomplete discourse in dialogs. In addition, our experimental results on the IUR task show that our IUR model greatly outperforms baseline and related systems, including large language models.

## Limitation

In this study, we investigate the effects of utterance rewriting methods on the enhancement of retrieval generation. Our research is centered on enhancing the generation capabilities of LLMs within RAG systems by optimizing retrieval effects. While the incorporation of IUR significantly enhances the quality of both retrieval and generation compared to standard retrieval methods, our exploration of the impact of retrieval results on generation remains incomplete. During the experiment, we find that not all queries positively influence the generation process of large models. In fact, some incorrect queries will mislead the large model generation. Moving forward, we will diligently examine how retrieval results impact models within RAG systems. By considering the necessity of retrieval, the relevance of retrieval outcomes, and other factors, we will further delve into the influence and optimization strategies for retrieval for LLMs and RAG systems.

## References


Vaibhav Adlakha, Shehzaad Dhuliawala, Kaheer Suleman, Harm de Vries, and Siva Reddy. 2022. TopiOCQA: Open-domain conversational question answering with topic switching. volume 10, pages 468–483.

Akari Asai, Sewon Min, Zexuan Zhong, and Danqi Chen. 2023a. Retrieval-based language models and applications. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 6: Tutorial Abstracts)*, pages 41–46.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023b. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729.

DeepSeek-AI. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

Ma Dongbo, Sami Miniaoui, Li Fen, Sara A Althubiti, and Theyab R Alsenani. 2023. Intelligent chatbot interaction system capable for sentimental analysis using hybrid machine learning algorithms. *Information Processing & Management*, 60(5):103440.

T Dozat. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Haowei Du, Dinghao Zhang, Chen Li, Yang Li, and Dongyan Zhao. 2023. Multi-granularity information interaction framework for incomplete utterance rewriting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2576–2581.

Ahmed Elgohary, Denis Peskov, and Jordan Boyd-Graber. 2019. Can you unpack that? learning to rewrite questions-in-context. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5918–5924.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023a. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6465–6488.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Xinnan Guo, Qian Zhu, Qiuhui Shi, Xuan Lin, Liubin Wang, DaqianLi DaqianLi, and Yongrui Chen. 2024. Context-aware tracking and dynamic introduction for incomplete utterance rewriting in extended multi-turn dialogues. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 2138–2148.

Shailja Gupta, Rajesh Ranjan, and Surya Narayan Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Jie Hao, Linfeng Song, Liwei Wang, Kun Xu, Zhaopeng Tu, and Dong Yu. 2021. Rast: Domain-robust dialogue rewriting as sequence tagging. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4913–4924.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong-Cheol Park. 2024. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. In *2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 7036–7050. Association for Computational Linguistics.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992.

Lisa Jin, Linfeng Song, Lifeng Jin, Dong Yu, and Daniel Gildea. 2022. Hierarchical context tagging for utterance rewriting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10849–10857.

Vineet Kumar and Sachindra Joshi. 2016. Non-sentential question resolution using sequence to sequence learning. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2022–2031.

Vineet Kumar and Sachindra Joshi. 2017. Incomplete follow-up question resolution using retrieval based sequence to sequence learning. In *Proceedings of the 40th international acm sigir conference on research and development in information retrieval*, pages 705–714.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Jiang Li, Xiangdong Su, Xinlan Ma, and Guanglai Gao. 2023a. How well apply simple mlp to incomplete utterance rewriting? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1567–1576.

Zitong Li, Jiawei Li, Haifeng Tang, Kenny Zhu, and Ruolan Yang. 2023b. Incomplete utterance rewriting by a two-phase locate-and-fill regime. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2731–2745.

Zuchao Li, Shitou Zhang, Hai Zhao, Yifei Yang, and Dongjie Yang. 2023c. Batgpt: A bidirectional autoregessive talker from generative pre-trained transformer. *arXiv preprint arXiv:2307.00360*.

Nelson F Liu, Tianyi Zhang, and Percy Liang. 2023a. Evaluating verifiability in generative search engines. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7001–7025.

Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020a. Incomplete utterance rewriting as semantic segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2846–2857.

Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020b. Incomplete utterance rewriting as semantic segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2846–2857, Online. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023b. Gpt understands, too. *AI Open*.

Zihan Liu, Wei Ping, Rajarshi Roy, Peng Xu, Chankyu Lee, Mohammad Shoeybi, and Bryan Catanzaro. 2024. Chatqa: Surpassing gpt-4 on conversational qa and rag. *Advances in Neural Information Processing Systems*, 37:15416–15459.

Hongyin Luo, Yung-Sung Chuang, Yuan Gong, Tianhua Zhang, Yoon Kim, Xixin Wu, Danny Fox, Helen Meng, and James Glass. 2023. Sail: Search-augmented instruction learning. *arXiv preprint arXiv:2305.15225*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822.

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

Zhufeng Pan, Kun Bai, Yan Wang, Lianqiang Zhou, and Xiaojiang Liu. 2019. Improving open-domain dialogue systems via multi-turn incomplete utterance restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1824–1833, Hong Kong, China. Association for Computational Linguistics.

Letian Peng, Zuchao Li, and Hai Zhao. 2024. Fast and accurate incomplete utterance rewriting. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W. Bruce Croft, and Mohit Iyyer. 2020. Open-Retrieval Conversational Question Answering. In *SIGIR*.

Jun Quan, Deyi Xiong, Bonnie Webber, and Changjian Hu. 2019. GECOR: An end-to-end generative ellipsis and co-reference resolution model for task-oriented dialogue. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4547–4557, Hong Kong, China. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Hui Su, Xiaoyu Shen, Rongzhi Zhang, Fei Sun, Pengwei Hu, Cheng Niu, and Jie Zhou. 2019. Improving multi-turn dialogue modelling with utterance rewriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2692–2700.

Shenzhi Wang, Yaowei Zheng, Guoyin Wang, Shiji Song, and Gao Huang. 2024. Llama3-8b-chinese-chat (revision 6622a23).

Kun Xu, Haochen Tan, Linfeng Song, Han Wu, Haisong Zhang, Linqi Song, and Dong Yu. 2020. Semantic role labeling guided multi-turn dialogue rewriter. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6632–6639.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Yao Yao, Zuchao Li, and Hai Zhao. 2024. Sirllm: Streaming infinite retentive llm. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2611–2624.

Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. 2023. Augmentation-adapted retriever improves generalization of language models as generic plug-in. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Yong Zhang, Zhitao Li, Jianzong Wang, Ning Cheng, and Jing Xiao. 2022. Self-attention for incomplete utterance rewriting. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8047–8051. IEEE.

ZhipuAI. 2024. Glmrag. Accessed: 2024-9-10.

## A RAG Experiments on Other Base Model

The experiment results on Qwen2.5-7B-Instruct (Team, 2024) and Llama3-8B-Chinese-Chat (Wang et al., 2024) are shown in Table 7. We are surprised to find that the IUR results on Qwen and Llama models are not ideal compared with ChatGLM3-6B (GLM et al., 2024). We check and find that this is due to the problem of rewriting the previous reference. Although this helps the model understand the question better, it can also lead to inconsistent responses, especially when the question answering of our dataset relies more on external data. The rewritten question is also incorrectly answered due to the lack of external knowledge, and the rewriting breaks the original dialogue coherence, so it is not effective. However, it can be found that the RAG system with IUR still has better results, which benefits from the enhancement of retrieval in the RAG process by IUR.

## B Encoder Setting

In BiLSTM sequence encoder setting, we use randomly initialized word embedding with dimension 100, character embedding with dimension 50, and set the character convolutional hidden size to 100. The embedding dropout is set to 0.2. Three BiLSTM layers are stacked with hidden size 600 and dropout 0.33. The hidden size of FFN for head and tail node projection is set to 600. While for the Transformer encoder setting, we fix the parameters of the bottom Transformer layers and only finetune the top 4 layers. We use a mean pooling strategy to deal with the inconsistency between the word-level parsing and subword-level representations. The hidden size of FFN is kept the same as in the BiLSTM encoder setting.

Our models are trained using Adam optimizer for up to 200 epochs, with early stopping based on development dataset performance. The learning rate is set to 1e-3 and 5e-5 for the BiLSTM encoder and Transformer encoder settings respectively. For English, we use the official BERT-large-uncased, and for Chinese, we use `BERT-base-chinese`.

## C Case Study of Dialogue-RAG

To further explore the enhancement effect of IUR on RAG, we conduct a case study. Figure 5 illustrates a dialogue sample in our Dialogue-RAG-MULTI-v1.0 dataset and Figure 6 shows its related document. Figure 7 and Figure 8, respectively, show the query results and responses enhanced without or with IUR. Through comparison, it can be found that the IUR enhanced question carry more critical information. This enhances the retrieval accuracy, which in turn enhances the RAG system's response accuracy.


```
{
  "lang": "en",
  "dialogue": [
    {
      "user": "What is the Athar Jamad Mosque known for?",
      "assistant": "It is one of the oldest and biggest mosques in Coimbatore."
    },
    {
      "user": "Who built the mosque?",
      "assistant": "A perfume merchant named Athar built it."
    },
    {
      "user": "How long did the construction take?",
      "assistant": "It took 44 years to complete."
    },
    {
      "user": "What architectural materials were used?",
      "assistant": "The structure is made of limestone and mortar."
    },
    {
      "user": "What special events attract crowds there?",
      "assistant": "Ramadan and Bakrid attract many visitors."
    },
    {
      "user": "Is there anything significant beside the mosque?",
      "assistant": "Yes, there is a dargah for Pir Jamesha Waliullah nearby."
    }
  ]
}
```


Figure 5: Sample Dialogue.


```
{
  "doc id": 7088,
  "doc content": "\nAthar Jamad Mosque\n\nMosque in Tamil Nadu, India\nAthar Jamad Masjid ( ; ; also known as \"The Big Mosque\" ) is located at Oppanakara Street within the Coimbatore district of Tamil Nadu in India. It is one of the oldest and biggest masjid in Coimbatore.\nHistory and construction.\nA perfume merchant named Athar, who migrated from Tirunelveli, built the structure. It took 44 years to complete, with construction beginning in 1860 and finishing in 1904. \nThe structure is made up of limestone and mortar and polished with egg white. The facade is covered with cusped arches surrounding the open courtyard, where the prayer halls stand. There is a covered ablution pond in the southeastern corner and a small library on the eastern side. There\'s also a kitchen that prepares \"nonbu kanji\" (rice soup) in the fasting period during Ramadan. Hawkers line the entrance with amulets and items of worship.\nAccording to the Indian National Trust for Art and Cultural Heritage (INTACH), the two minars with domed roofs on the northern and southern sides are 85 feet high. This pair of silver domes stand out in the skyline of the Town Hall area. The mosque can accommodate about 2,000 worshippers during Friday prayers.\nDargah.\nThe mosque is built beside the tomb of Pir Jamesha Waliullah, a Sufi Waliullah who died in the 1850s. His tomb, which is now a dargah in the middle of Big Bazaar Street, is on the southern side of the masjid. Visitors are blessed inside the dargah with amulets tied around their necks to ward off evil spirits.\nFollowing.\nThe Jamaat comprises the descendants of the 52 families from Tirunelveli that moved to Coimbatore in 1850. According to Jamaat secretary A.R. Baserdeen, 1355 members are now alive. \nThe Jamaat\'s elected executive committee manages the mosque, as well as Jamesha Waliullah dargah on Big Bazaar Street, Jungal Pir dargah on Trichy road, and the Cemetery Masjid beside Coimbatore Junction. The committee also runs three schools in the area which serve 1200 students.\nDuring Ramadan and Bakrid, crowds flock to the masjid and the dargah beside it.\nReferences.\n\n\n"
}
```


Figure 6: Sample Document.

## D Supplementary Dataset Experiments for IUR Task

We conducted additional comparative experiments on English Task-Oriented Dialogue dataset MULTI (Pan et al., 2019) and Chinese Open Domain Dialogue dataset TASK (Quan et al., 2019).

| Method | | $\mathbf{B}_1$ | $\mathbf{B}_2$ | $\mathbf{B}_3$ | $\mathbf{B}_4$ | $\mathcal{R}_1$ | $\mathcal{R}_2$ | $\mathcal{R}_L$ | $\mathcal{S}_p$ | $\mathcal{S}_r$ | $\mathcal{S}_f$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Qwen2.5 | w/o IUR | 9.04 | 5.54 | 3.90 | 2.77 | 14.97 | 5.79 | 13.96 | 58.43 | 63.53 | 60.69 |
| | w/ IUR | 9.32 | 5.49 | 3.72 | 2.51 | 15.43 | 5.69 | 14.27 | 57.83 | 65.86 | 61.44 |
| | RAG w/o IUR | 10.61 | 7.45 | 5.73 | 4.39 | 20.75 | 10.96 | 19.54 | 59.22 | 70.02 | 63.96 |
| | **Dialogue-RAG (ours)** | **14.20** | **10.81** | **8.62** | **6.76** | **29.27** | **17.07** | **27.8** | **62.39** | **77.39** | **68.9** |
| LLaMA3 | w/o IUR | 7.79 | 4.71 | 3.29 | 2.32 | 13.11 | 5.05 | 12.11 | 54.7 | 61.75 | 57.82 |
| | w/ IUR | 7.74 | 4.55 | 3.08 | 2.06 | 13.54 | 4.95 | 12.4 | 54.4 | 63.77 | 58.56 |
| | RAG w/o IUR | 9.76 | 6.92 | 5.37 | 4.14 | 19.10 | 10.15 | 18.07 | 56.58 | 68.47 | 61.69 |
| | **Dialogue-RAG (ours)** | **13.00** | **9.80** | **7.78** | **6.09** | **26.82** | **15.59** | **25.44** | **60.28** | **76.02** | **66.99** |

Table 7: RAG experiments on Qwen2.5 and LLaMA3. $B$, $R$, and $S$ are BLEU, ROUGE, and BERT SCORE respectively. The cumulative n-gram BLEU score is denote as $\boldsymbol{B_n}$. For ROUGE metric, it measures the n-gram overlapping (denoted as $\boldsymbol{R_n}$) and longest matching (denoted as $\boldsymbol{R_L}$) between the rewritten utterances and the golden ones. $\mathcal{S}_p$, $\mathcal{S}_r$, and $\mathcal{S}_f$ represent the precision, recall, and F1 score of BERT SCORE respectively.


```
{
    "question": "How long did the construction take?",
    "ground truth": "It took 44 years to complete.",
    "retrieval results": [
        {"chunk id": 18941, "doc id": 5462, "content": " The work was suspended
several times due to violation of the contract clauses by the customer. As of
2024, the work is not completed.\nReferences.\n\n\n"},
        {"chunk id": 23971, "doc id": 6828, "content": "On March 6 & 10, 2011,
installation of the 653 ton ea. sector gates was completed.\nThe Notice of
Construction Complete was issued on April 8, 2014.\nSources.\n\n\n"},
        {"chunk id": 17610, "doc id": 5109, "content": "The first such conditions
were created by the Novi Sad Urban Institute in 1970. The Novi Sad Municipal
Assembly decides on the construction of the bridge in December 1973. The
architects of the bridge were academics Nikola Hajdin, Gojko Nenadić and
Predrag Želalić.\nWork began on 22 September 1976, when the first pier on the
left bank of the Danube was stabbed. During the construction, the attention of
the experts was focused on three details: The construction of the bars for pillar
17, where there were traces of land slides; the assembly of the steel structure, in
pieces, weighing 200 tonnes, without scaffolding, with the help of the Derrick
cranes; and the assembly of the steel ropes, which was original.\n"}],
    "model response": "The construction of the bridge took approximately 48
months."
}
```


Figure 7: The case of RAG result without IUR.


```
{
    "question": "How long did the construction take?",
    "ground truth": "It took 44 years to complete.",
    "rewritten question": "How long did the construction of the Athar Jamad Mosque take?",
    "retrieval results": [
        {"chunk_id": 24918, "file_id": 7088, "content": "\nAthar Jamad Mosque\n\nMosque in Tamil
Nadu, India\nAthar Jamad Masjid ( ; ; also known as \"The Big Mosque\" ) is located at Oppanakara
Street within the Coimbatore district of Tamil Nadu in India. It is one of the oldest and biggest masjid
in Coimbatore.\nHistory and construction.\nA perfume merchant named Athar, who migrated from
Tirunelveli, built the structure. It took 44 years to complete, with construction beginning in 1860 and
finishing in 1904. \nThe structure is made up of limestone and mortar and polished with egg white."},
        {"chunk_id": 30564, "file_id": 8609, "content": " The analysis of the brickwork during the
restoration showed different construction phases, and revealed that the foundations and the
surviving walls were made of brick and stone. Moreover, the surveys show that during the
conversion into a mosque in 1506 the atrium and the wall of the Mihrab had to be rebuilt. At the
same time, a massive minaret was erected on the northeast side of the
building.\nReferences.\n\n\n"},
        {"chunk_id": 16735, "file_id": 4877, "content": "Construction of the mosque took four years. It is
10,000 square meters in area. It has two minarets, each 35 meters high, and a dome of aluminum
composite. The calligraphy on the interior of the dome was written by Semih İrteş. The large
windows on three sides of the prayer hall were designed by Orhan Koçan. The minbar is acrylic and
was designed by Tayfun Erdoğmuş. Decorative motifs are derived from Seljuk art. The large,
asymmetrical chandelier has waterdrop-shaped glass globes made by Nahide Büyükkaymakçı,
\"reflecting a prayer that Allah's light should fall on worshipers like rain,\" and the women's section is
designed especially to allow a clear view of the chandelier."}],
    "model response": " The construction of the Athar Jamad Mosque in Coimbatore, Tamil Nadu,
India took 44 years, from 1860 to 1904."
}
```


Figure 8: The case of RAG result with IUR.

### D.1 Results on Multi Dataset

In Table 8, we report the IUR performance on the MULTI dataset and list five baselines for comparison: Syntactic, L-Gen, L-Ptr-Gen, PAC, and RUN.

Our method formulates IUR as a digraph parsing problem and proposes CSF and RCF features to strengthen the parsing process, thus achieving better IUR performance than the previous best system, improving on $\mathcal{F}_1$, $\mathcal{F}_2$, and $\mathcal{F}_3$ metrics, 2.4, 1.9, 2.0, respectively. For the final generated rewrite utterance, we get an improvement in BLEU and ROUGE metrics, reaching +0.8 $\mathbf{B}_1$, +0.8 $\mathbf{B}_2$, +0.5 $\mathbf{R}_1$, and +1.3 $\mathbf{R}_2$. On one hand, these improvements are due to digraph modeling and node linking parser, which can seek the insertion span and the insertion position more easily than UNet, due to only single-step digraph derivation is required, rather than enumerating all the words in the span as in RUN. On the other hand, the features derived from our generative PrLM can effectively help remove unreasonable digraph links, reducing the search space for insertion decisions.

In the case of using BERT PrLM, PAC (Pan et al., 2019) proposes a "pick-and-combine" model that demonstrates strong competitiveness. We also achieved good improvements within this setting, with $\mathcal{F}_1$ +2.1, $\mathcal{F}_2$ +2.0, $\mathcal{F}_3$ +1.3 scores respectively, and 0.8, 1.2, 0.6, 1.2 gains on $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{R}_1$, $\mathbf{R}_2$ compared to RUN. This further illustrates the advantages of our digraph modeling compared to formalization as semantic segmentation. In addition, the improvements on $\mathbf{B}_1$, $\mathbf{B}_2$, $\mathbf{R}_1$, and $\mathbf{R}_2$ show that our method makes the determination of context span and insertion point more accurate.

### D.2 Results on Task Dataset

Experiment results on the TASK benchmarks are presented in Table 9. We have additionally introduced two systems for comparison, GECOR and Pronoun Sub, of which GECOR 1 adopts the copy mechanism, while GECOR 2 adopts the gated copy mechanism. For a fair comparison, experiments

| Model | $\mathcal{P}_1$ | $\mathcal{R}_1$ | $\mathcal{F}_1$ | $\mathcal{P}_2$ | $\mathcal{R}_2$ | $\mathcal{F}_2$ | $\mathcal{P}_3$ | $\mathcal{R}_3$ | $\mathcal{F}_3$ | $\mathbf{B}_1$ | $\mathbf{B}_2$ | $\mathbf{R}_1$ | $\mathbf{R}_2$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DuS † | 67.4 | 37.2 | 47.9 | 53.9 | 30.3 | 38.8 | 45.3 | 25.3 | 32.5 | 84.1 | 81.2 | 89.3 | 80.6 |
| L-Gen † | 65.5 | 40.8 | 50.3 | 52.2 | 32.6 | 40.1 | 43.6 | 27.0 | 33.4 | 84.9 | 81.7 | 88.8 | 80.3 |
| L-Ptr-Gen † | 66.6 | 40.4 | 50.3 | 54.0 | 33.1 | 41.1 | 45.9 | 28.1 | 34.9 | 84.7 | 81.7 | 89.0 | 80.9 |
| RUN (Liu et al., 2020b) | 66.9 | 54.9 | 60.3 | 53.0 | 43.4 | 47.7 | 43.8 | 35.7 | 39.3 | 91.1 | 88.0 | 91.0 | 83.3 |
| **Ours** | 67.3 | 58.7 | **62.7** | 53.7 | 46.1 | **49.6** | 44.9 | 38.3 | **41.3** | **91.9** | **88.8** | **91.5** | **84.6** |
| PAC † | 70.5 | 58.1 | 63.7 | 55.4 | 45.1 | 49.7 | 45.2 | 36.6 | 40.4 | 89.9 | 86.3 | 91.6 | 82.8 |
| RUN + BERT | 73.2 | 64.6 | 68.6 | 59.5 | 53.0 | 56.0 | 50.7 | 45.1 | 47.7 | 92.3 | 89.6 | 92.4 | 85.1 |
| **Ours + BERT** | 74.5 | 67.4 | **70.7** | 60.8 | 55.4 | **58.0** | 51.2 | 46.4 | **48.7** | **93.1** | **90.8** | **93.0** | **86.3** |

Table 8: Experimental results on the MULTI dataset. †: Results from Pan et al. (2019). The bolded results in a column indicate a statistically significant improvement against all the baselines ($p < 0.05$).

| **Model** | **EM** | $\mathbf{B}_4$ | $\mathcal{F}_1$ |
|---|---|---|---|
| L-Ptr-Gen † | 50.4 | 74.1 | 44.1 |
| GECOR 1 † | 68.5 | 83.9 | 66.1 |
| GECOR 2 † | 66.2 | 83.0 | 66.2 |
| RUN (Liu et al., 2020b) | 69.2 | 85.6 | 70.6 |
| **Ours** | 71.2 | 86.7 | 72.8 |

Table 9: Experimental results on TASK. †: Results from Quan et al. (2019).

are not augmented with BERT. From the results, our model obtains the best results on EM, BLEU, and ROUGE. In conclusion, on four datasets with different languages and different domains, these improvements illustrate digraph modeling, CSF and RCF features are generally effective for IUR.

## E Case Study of Rewrite

Table 10 shows a specific case-based comparison of performance between our model and baseline RUN. To rewrite the final utterance *Could you give me the phone number and postcode ?*, our model can locate the insertion relationship clearly, and select the appropriate spans *one restaurant listing for North American food* and *the only restaurant serving north American food* for action. In contrast, RUN resolves the reference to *north American food* wrongly from the previous context. Though *north American food* forms a proper supplementary for the sentence, it is meaningless due to the user asks for *restaurant* instead of *north American food*. The possible reason is that the semantic segmentation and region searching algorithms are misled by the inference error and decode a grammatically correct but inappropriate span for insertion. The error from CNNs prediction in RUN is passed to the decoding algorithm, resulting in such inconsistency with the fact, which makes the region wrongly attended and weakens the model performance. Thus, it demonstrates that our model shows better robustness and is free from disturbing factors compared with region-based RUN.

In addition, comparing the model outputs with and without CSF and RCF features, although both our models found the correct reference, i.e. *American food restaurant*, the one without CSF and RCF features chooses Utterance 4, while w/ CSF,RCF finds Utterance 5. Since CSF and RCF bring constraints from contextual similarity and rewriting consistency, *the only restaurant serving American food* in Utterance 5 is more consistent with the requirement. This also illustrates the effectiveness of our proposed CSF and RCF for IUR. However, in the final generation from surface tokens, there is an additional *only* compared to a gold reference, which is a common problem in extractive IUR modeling, but it does not have much effect on the actual role of IUR.

| |
|---|
| **Utterance 1:** I am looking for a Danish restaurant .<br>**Utterance 2:** There are no danish restaurants listed . may i direct you toward another restaurant ?<br>**Utterance 3:** How about American food ?<br>**Utterance 4:** There is one restaurant listing for north American food , gourmet burger kitchen in centre part of town .<br>**Utterance 5:** Is there any north American food in centre part of town else ?<br>**Utterance 6:** That is the only restaurant serving north American food .<br>**Utterance 7:** Could you give me the phone number and postcode ? |
| **Gold:** Could you give me the phone number and postcode of the restaurant serving north American food? |
| **RUN:** Could you give me the phone number and postcode of north American food ? |
| **Ours:** Could you give me the phone number and postcode of one restaurant listing for north American food? |
| **Ours w/ CSF,RCF:** Could you give me the phone number and postcode of the only restaurant serving north American food? |

Table 10: Case study for comparison between our model and baselines.